Self-Supervised Deep Metric Learning for Pointsets

Pattaramanee Arsomngern[1], Cheng Long[2], Supasorn Suwajanakorn[3], Sarana Nutanong[4]
[1,3,4]*School of Information Science and Technology, Vidyasirimedhi Institute of Science and Technology, Thailand*
[2]*School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[1,3,4]{pattaramanee.a_s19,supasorn.s,snutanon}@vistec.ac.th, [2]c.long@ntu.edu.sg

***Abstract*—Deep metric learning is a supervised learning paradigm to construct a meaningful vector space to represent complex objects. A successful application of deep metric learning to pointsets means that we can avoid expensive retrieval operations on objects such as documents and can significantly facilitate many machine learning and data mining tasks involving pointsets. We propose a self-supervised deep metric learning solution for pointsets. The novelty of our proposed solution lies in a self-supervision mechanism, that makes use of a distribution distance for set ranking called the Earth's Mover Distance (EMD) to generate pseudo labels. Our experimental studies on four documents datasets show that our proposed solution outperform baselines and state-of-the-art approaches on unsupervised deep metric learning in most settings.**

***Index Terms*—set retrieval, deep metric learning, self-supervised learning, triplet loss, earth mover's distance**

I. Introduction

A pointset, in the form of a set of vectors or points, can represent unstructured data such as documents [1] and video clips. We can retrieve documents most similar to a query document effectively using a distribution distance, such as the Earth Mover's Distance (EMD) [2] as the ranking function. However, its computational complexity, which is $O(n^3 \log(n))$, may be prohibitive for a real-time retrieval system, where n is the number of elements in the set. One way to mitigate this problem is to construct an entirely new L_p norm embedding vector space. This embedding idea has been adopted for various applications including image retrieval [3], face recognition [4], [5], person re-identification [6], and trajectory similarity search [7], [8].

One main limitation of deep metric learning is that it requires supervision, which makes it impossible to use with unlabeled datasets or hard to generalize when the number of labels is small. A popular approach to overcome this challenge is to apply unsupervised methods. However, these techniques are designed specifically for computer vision [9], [10] or natural language processing (NLP) [11] rather than generic pointsets.

In this paper, we propose a generic solution for pointset metric learning that can alleviate learning challenges on unlabeled or small datasets. The crux of our proposed solution lies in a new loss function called *Weighted Self-supervised EMD Triplet (WSSET)*, which enables embedding space training without labels. Our solution constructs a vector space in which the Euclidean distance between vectors resembles the relationship between their pointsets in the EMD space and provides better performance than a straightforward method that tries to approximate the EMD directly. The vector space constructed by our solution also provides the simplicity of distance calculation and indexing. Furthermore, we propose sequential transfer learning strategies that can transfer the learned relationships as prior knowledge for improving the performance of learning a target problem.

We conducted extensive experimental studies on pointset documents [1] to evaluate against well-known baselines and state-of-the-art supervised and unsupervised techniques. Results show that our proposed solution outperforms the competitors in most settings and works exceptionally well in simulated scenarios when the training data is limited (i.e., small dataset conditions). In addition, our method also has lower computation time than other competitors.

The contributions of our work are as follows. First, we propose a novel deep metric learning loss function based on EMD and a learning method to enable self-supervised training on any unlabeled pointsets. Second, we introduce a transfer learning strategy to overcome the small dataset conditions. Third, we avoid the expensive computational complexity of EMD calculation by formulating an embedding function that maps pointsets into the Euclidean space. Fourth, we conducted extensive experiments to compare our proposed solution against state-of-the-art techniques for both unsupervised and supervised learning approaches.

II. Related Work

A. Pointsets Similarity Retrieval

The main challenge of finding similar pointsets lies in the cost of distance calculation. For example, using the Earth Mover's Distance (EMD) [2], which is a well-known measure for pointsets, we end up with a cubic (or cubic-logarithmic) complexity with respect to the number of elements per distance calculation. [12] improves the efficiency of EMD by using approximation techniques. However, it still suffers from issues of computational cost and indexing [13]. In the next subsection, we will discuss how deep metric learning can address these issues.

B. General Deep Metric Learning

Deep metric learning is a popular technique to learn an embedding function in a supervised learning manner, such that the distances between intra-class samples are significantly smaller than those between inter-class samples. There are two main methods for embedding learning: *pair-based* and *triplet-based*. Results from various experiments [14] show that using a shared reference point in the triplet-based methods provide an improvement over the pair-based ones. Also, focusing on

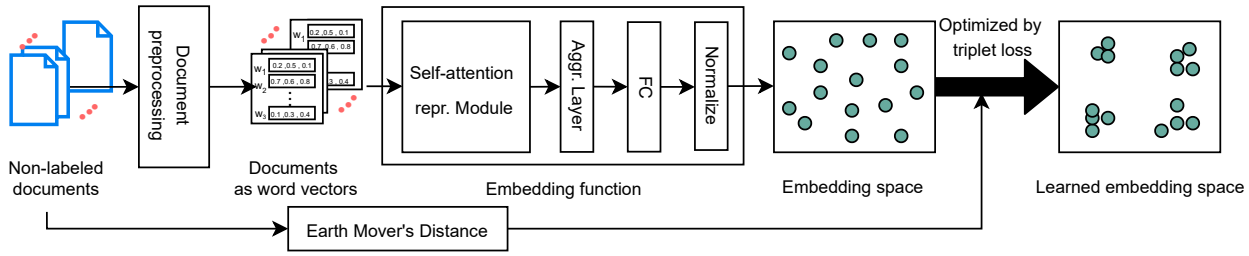

Fig. 1. The overall architecture of our proposed method using document input for illustrative purposes. We first process each unlabeled document into a set of word vectors, then embed this set into our embedding space with a permutation invariant neural network. This embedding network is optimized using our triplet loss in a self-supervised fashion through our automatic triplet selection.

critical negatives in the triplet-based methods outperform a baseline that chooses triplets at random [4], [6]. In our work, we employ the triplet loss with a semi-hard negative sampling [4] and propose a self-supervised method to train it without labels, which were traditionally required.

C. Unsupervised and Self-supervised Deep Metric Learning

Recently, unsupervised and self-supervised embedding learning receive growing research attention, especially in the field of computer vision. [9], [10] formulate a self-prediction problem in which each instance is its own distinct class by using a CNN with contrastive learning.

Alternatively, [15] propose a method which utilizes a pre-trianed *ImageNet* model to provide an intermediate embedded space for mining triplets.

These mentioned techniques work well for image retrieval tasks without using labels. Unlike these studies, we propose a novel self-supervised deep metric learning technique for embedding the unlabeled pointsets, which is *not* domain-specific.

D. Deep Learning on Pointsets

In our work, we make use of deep learning on pointsets. One of the first challenges of applying a neural network to process pointsets is the mismatch between the order invariance of pointsets and the order sensitivity of the network's input. An early attempt to address this challenge provides a specific convention to reorder the elements before putting them into an LSTM network [16]. Recent approaches introduce permutation invariance directly into the model, e.g., DeepSets [17], and another area of research also aims to learn pointsets representation in an unsupervised fashion, i.e., set autoencoder [18].

One advancement in deep learning on pointsets is the application of self-attention mechanism i.e., Transformer [19], which helps capture the relationship between set elements [20]. In our study, we apply the self-attention mechanism by combining Transformer and DeepSets for encoding pointsets and learning a representation in a self-supervised way.

III. Problem Formulation

Consider three pointsets x_i, x_j, and x_k represented as sets of vectors. Let $S(,)$ and $f()$ be a set similarity function and an embedding function for pointsets, respectively. Our goal is to learn $f()$ such that if $S(x_i, x_j) < S(x_i, x_k)$, then $||f(x_i) - f(x_j)|| < ||f(x_i) - f(x_k)||$. Specifically, the output embedding $f(x) \in \mathbb{R}^F$ is located in an embedding space associated with the Euclidean metric. Given this embedding space, one can solve a retrieval task or classify a query input with, e.g., k-nearest neighbor algorithm. The main challenge is how to find f that well generalizes to downstream tasks from *zero training labels*.

IV. Proposed Solution

In this section, we describe our proposed solution in detail. For ease of exposition, we use the application of document classification as our narration framework. However, our solution can be applied to any classification problem in which data entries can be represented as pointsets.

Our key idea is to parameterize the embedding function as a permutation invariant neural network f_θ and use a triplet loss [4] with our *pseudo* labels to optimize for the embedding space. In triplet loss optimization with supervision, an "anchor" document will be forced closer to a document belonging to the same class and farther away from a document of a different class, called the positive and negative respectively. Our key contribution is a self-supervised method for identifying the positive and negative pairs associated with each anchor document without ground-truth labels. The overview of our proposed solution is shown in Fig. 1.

Inspired by the success of Earth mover's distance (EMD) [1] used for document classification, image and multimedia retrieval tasks, we propose to use EMD for inferring the positive and negative pairs. In particular, a document with a low EMD to an anchor document will be labeled positive, and a document with a high EMD will be labeled negative. Moreover, we also introduce a re-weighting approach that places more emphasis on negative documents that are more likely to belong to different classes. For the embedding function, we propose a neural network based on a state-of-the-art architecture in NLP, i.e., Transformer [19] and its self-attention mechanism that achieves permutation invariance suitable for representing documents.

A. Self-supervised EMD triplet loss

To train our network with a triplet loss, we first need to form a set of document triplets, each of which consists of an anchor, a positive, and a negative as shown in Fig. 2. Given a mini-batch of size n, we generate n training triplets by taking

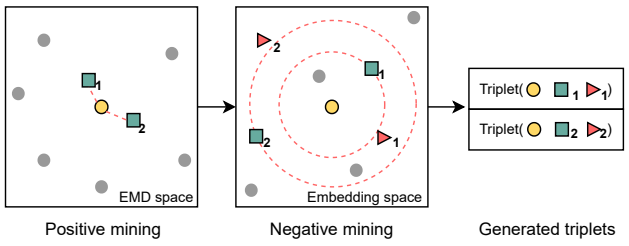

Fig. 2. An illustration of triplet selection. Given an anchor point (yellow circle), the closest point to this anchor in EMD space is chosen as the positive (green rectangle). Then, the negative is the closest point to the anchor that is at least as far way as the distance between the anchor and the positive (dotted ring) in the embedding space.

each document x_i in the mini-batch as the anchor and taking the closest document to each anchor in the EMD space as the positive x_i^p. For the negative, we follow the semi-hard negative mining strategy from [4]. This strategy mines a negative based on document embedding from $f()$ by choosing a document closest to the anchor in the embedding space but is farther away from the anchor than the positive. Formally, the negative x_i^n is a document closest to the anchor in the embedding space that satisfies: $\left\|f(x_i^a) - f(x_i^p)\right\|_2^2 < \left\|f(x_i^a) - f(x_i^n)\right\|_2^2$. With these training triplets, we then train our mapping function, parametrized by a neural network f with the following triplet loss:

$$L(X) = \frac{1}{\hat{n}} \sum_{i=0}^{\hat{n}} \left[\left\|f(x_i^a) - f(x_i^p)\right\|_2^2 - \left\|f(x_i^a) - f(x_i^n)\right\|_2^2 + \alpha \right]_+, \tag{1}$$

where X is the mini-batch, $\hat{n}$ is the total number of triplets in each mini-batch, and α is a hyper-parameter denoting the desired margin between the anchor-positive and anchor-negative distances in the embedding space.

We call this loss *self-supervised EMD triplet loss* or *SSET*. Minimizing this loss encourages neighbor documents in the EMD space, potentially belonging to the same class, to be close to one another in the embedding space, as well as maximizing the distances of documents in the embedding space whose EMDs are high.

It is interesting to note that we can approximate EMD among items directly using regression loss (i.e., $L(X) = \frac{1}{n^2}\sum_{i,j}^{n}(\text{EMD}(x_i, x_j) - \parallel f(x_i) - f(x_j) \parallel_2^2)^2$). Empirically, we found that our proposed triplet loss is easier to train and consistently gives better results than the direct EMD approximation loss (See Section VI-B).

B. Re-weight the negative pair using EMD

One of the challenges of semi-hard negative sampling without supervision is that we cannot be certain that those negative documents that are quite close to the positives truly belong to different classes. This could be especially problematic in our self-supervised setting. One way to alleviate the effect of mislabeling is to prioritize negative documents that are more likely to come from different classes, which can be captured by higher EMDs to their anchors [1]. To achieve this, rather than using a hard negative sampling strategy, which could lead to minimal learning [4], we simply re-weight the negative pair distance so that the more likely negatives are moved farther away than the less likely ones. In particular, our modified triplet loss has the following form:

$$L(X) = \frac{1}{\hat{n}} \sum_{i=0}^{\hat{n}} \left[\left\|f(x_i^a) - f(x_i^p)\right\|_2^2 - w_i \left\|f(x_i^a) - f(x_i^n)\right\|_2^2 + \alpha \right]_+, \tag{2}$$

where w_i is an exponential decay re-weighting factor for the i^{th} triplet and is defined as

$$w_i = \exp\left(-\frac{EMD(x_i^a, x_i^n)}{2(c\sigma)^2}\right), \tag{3}$$

where σ is the standard deviation of all pairwise EMD distances in each mini-batch and c is a tunable scaling for σ. Negative documents whose EMDs are higher will be given lower w_i, and in order to satisfy the margin constrain of the triplet loss, the network needs to make these negatives even farther away from their anchors. We call this loss function *Weighted self-supervised triplet loss* or *WSSET*. Surprisingly, such a simple modification also consistently outperforms the standard version given in Eq. 1. Please refer to our experimental section for results.

C. Self-attention permutation invariance representation

To model the embedding function $f(x)$ in Eq. (2) as a permutation invariant neural network, we follow a general approach [17] that decomposes the final embedding of a set into a sum of individual embeddings of its elements. By the commutative property of addition or aggregation, the final embedding is thus permutation invariant. In our model, the embedding function that is applied to each individual element, a word vector w in our case, is modeled using a self-attention module [19], which we will refer to as *Encoder*. Our final embedding function is $f(x) = \rho\left(\sum_{w \in x} \text{ENCODER}(w)\right)$, where ρ is modeled using three fully-connected layers. Our self-attention module, *Encoder*, is a stack of Transformer's encoders [19] similar to the one proposed in a state-of-the-art model BERT [11] in NLP. These encoders help the network understand the inner-set correlation of the documents, and by stacking them multiple times, we allow the network to model more complex relationships between each word in the documents.

V. Application: Transfer Learning

In this section, we analyze the benefits of our proposed self-supervised learning loss. We hypothesize that the learned representation based on EMD space could be useful for learning a target task. To test this, we apply a "sequential" transfer learning method [21] by using our learned embedding function as an input feature extractor for the target problem. We introduce two strategies: *Internal-data transfer learning* (i.e., Ours+Int.) and *External-data transfer learning* (i.e.,

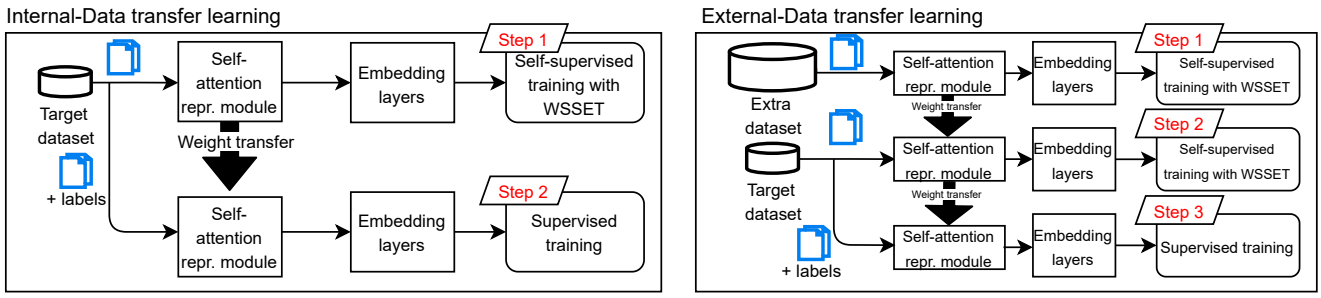

Fig. 3. Illustrations of two transfer learning strategies. The left figure shows internal-data transfer learning which first constructs a base model from the target dataset without using labels, before fine-tuning the model with labels. The right figure shows external-data transfer learning which first constructs a base model from an additional dataset without using labels, then followed by step 2 and 3 which are similar to the first two steps of internal-data transfer learning.

Ours+Ext.). The difference between these two strategies is in the way a base model is trained. Ours+Int. trains the base model using the target dataset only, while Ours+Ext. is using additional datasets as shown in Fig. 3.

With these ideas, we can formulate a set of hypotheses forming the basis for experimental studies in the next subsection.

1) We expect the learned representation based on our proposed loss to have comparable performance to the exact EMD method.
2) We expect that the two transfer learning strategies should outperform supervised learning baselines. The learned representation from the self-supervised learning stage could improve the performance of downstream task, especially, in small dataset conditions, by using the learned representation from additional datasets, i.e., external-data transfer learning.

VI. Experiments

To evaluate our proposed method and extension ideas from the previous section, we use four datasets for document retrieval and classification tasks [1], [22]: *BBCSports, Twitter, Recipe*, and *Amazon* with the original train/test splits. We use classification accuracy as the main evaluation metric similar to the practice in existing literature [1], [22] and Recall@K metric to measure the retrieval performance.

A. Implementation

Our *Encoder* in Section IV-C uses a stack of 5 Transformer's encoders, each of which uses 7 attention heads in the multi-head attention module and a feed-forward layer with 1,000 hidden nodes. The ρ function is modeled with three fully-connected layers of sizes 512, 256, and 64, where only the first two have ReLU activations. We also normalize the last layer using L2 normalization. The input document is converted into a set of word vectors using GoogleNews Word2Vec model with a feature size of 300. We also append the word frequency to the feature vector, resulting in our feature of size 301. Additionally, we zero-pad the dimension corresponding to the number of words in each document so that each document has the same number of words, i.e., the maximum number of words in all documents.

We trained our network with a batch size of 64 using Adam optimizer with a learning rate of 10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$ for 1,000 epochs for each experiment and use the model that has the best accuracy on the validation set for testing. For WSSET, we use $\alpha = 0.1$ and $c = 7$.

In the transfer learning strategies, we train a base model using the same WSSET setting. In particular, we used *Consumer Reviews of Amazon Products* dataset on Kaggle to train a base model in the external-data transfer learning model. For the supervised training step, we use a triplet loss with $\alpha = 0.1$ and the semi-hard negative mining strategy proposed in [4].

To benchmark our models, we use a distance-weighted k-nearest neighbor classifier (kNN) with $k = 10$ to predict labels on classification task experiments on Section VI-B, VI-C, and VI-E. [1]

For other competitors that are not designed for pointsets data, we use DeepSets [17] architecture to encode pointsets input and train the models using their proposed losses.

B. Comparison with state-of-the-art approaches

In this section, we evaluate retrieval and classification performance of our proposed WSSET loss and internal/external-data transfer learning model (i.e., Ours+Int. and Ours+Ext.) against state-of-the-art approaches.

For the classification task, Table I shows that our proposed WSSET loss, i.e., Ours, outperforms all unsupervised and self-supervised methods, including Exemplar [9], UDML-SS [15], and Approx.EMD. However, we perform better than the distribution distance method only on BBCSport (i.e., WMD). Nonetheless, WMD takes $O(n^3 \log n)$ for one distance calculation while our method has a much lower computation cost. Lastly, Approx.EMD, which is a direct EMD approximation method, has poorer performance than ours as mentioned in Section V but still outperforms the other two approaches.

In the supervised learning setting without using external training data, Ours+Int. dominates all competitors, i.e., DSPN [18] with a softmax layer, RepSet [22], DeepSets [17] with cross-entropy and triplet loss, and self-attention representation module (Section IV-C) with triplet loss. This result shows that the representation learned from EMD space by our proposed loss is useful for learning the target task.

[1] Code and data available at https://github.com/vistec-AI/WSSET/

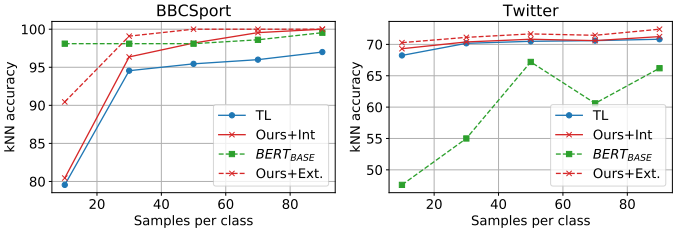

Fig. 4. This figure reports kNN accuracies for different methods explained in Section VI-C. We simulate scenarios when the training data is limited by reducing the number of samples per class in the dataset. Solid lines represent approaches that do not use any external data, while dashed lines represent approaches that do.

In the setting trained with external data, Ours+Ext. outperforms $\text{BERT}_{\text{BASE}}$ on all datasets except Recipe, which we did not use for comparison because Recipe lacks raw text input that BERT requires. Furthermore, Ours+Ext. also beats Ours+Int. on Recipe and Amazon datasets. This demonstrates that the prior knowledge learned from external datasets could guide the model to learn a target task better in the aforementioned datasets.

For the retrieval task, Table II also shows similar results. Ours outperforms all unsupervised methods, including WMD at lower K on BBCSports and Amazon datasets. Ours+Int. can outperform all supervised methods that do not use external data, while Ours+Ext. approach has comparable results to $\text{BERT}_{\text{BASE}}$. In addition, Ours+Ext. method always has better performance at lower K. This is due to the fact that a representation from the base model focuses on the relationship between the nearest sample (i.e., lower K) rather than the farther one (i.e., higher K).

TABLE I
COMPARISON OF CLASSIFICATION ACCURACIES WITH OTHER SUPERVISED/UNSUPERVISED APPROACHES (SUP) AND APPROACHES USING EXTERNAL DATA (EXT).

Methods	**Sup**	**Ext**	**Datasets**			
			BBCSport	***Recipe***	***Amazon***	***Twitter***
WMD	X	X	**95.0**	**58.42**	**92.75**	**71.49**
Exemplar	X	X	58.64	33.26	69.79	70.17
UDML-SS	X	X	60.9	28.60	72.45	65.66
Approx.EMD	X	X	93.18	50.13	84.79	70.49
Ours	X	X	**96.36**	**51.87**	**92.17**	**70.71**
DeepSets+CE	✓	X	97.7	58.88	94.95	75.42
DeepSets+TL	✓	X	99.09	54.3	94.10	73.6
SelfAttn+CE	✓	X	97.27	61.00	94.30	76.6
SelfAttn+TL	✓	X	97.75	62.30	94.65	77.5
RepSet	✓	X	98.00	61.43	94.71	74.58
DSPN+CE	✓	X	95.19	47.77	93.29	68.80
Ours+Int.	✓	X	**100.00**	**63.16**	**95.50**	**78.33**
$\text{BERT}_{\text{BASE}}$	✓	✓	99.54	-	95.21	75.85
Ours+Ext.	✓	✓	**100.00**	**63.31**	**95.63**	**78.11**

C. Experiments on small dataset conditions

This experiment demonstrates how our proposed transfer learning strategy can help alleviate small dataset limitations in some learning task. We generate multiple smaller versions of BBCsports and Twitter datasets with varying numbers of average samples per class ranging from 10 to 90 while retaining the original class distribution.

TABLE II
COMPARISON OF RECALL@K WITH OTHER APPROACHES

BBCSports	**Sup**	**Ext**	**R@5**	**R@15**	**R@30**	**R@45**
WMD	X	X	7.28	**24.31**	**46.98**	**66.79**
Exemplar	X	X	4.14	10.49	18.61	25.99
Approx.EMD	X	X	**8.01**	21.89	36.17	46.25
Ours	X	X	**9.10**	**25.64**	**45.59**	**58.97**
SelfAttn+TL	✓	X	9.44	28.45	56.16	72.78
DSPN+TL	✓	X	2.83	8.36	16.76	24.72
Ours+Int.	✓	X	**9.61**	**28.88**	**57.21**	**74.16**
$\text{BERT}_{\text{BASE}}$	✓	✓	9.69	29.05	**57.64**	**75.06**
Ours+Ext.	✓	✓	**9.70**	**29.07**	57.57	**75.06**
Recipe	**Sup**	**Ext**	**R@25**	**R@50**	**R@100**	**R@200**
WMD	X	X	**4.45**	**8.30**	**15.20**	**27.18**
Exemplar	X	X	1.92	3.89	7.69	15.43
Approx.EMD	X	X	3.54	6.42	11.54	20.75
Ours	X	X	**4.19**	**7.73**	**13.82**	**23.96**
SelfAttn+TL	✓	X	5.69	11.10	21.04	38.98
DSPN+TL	✓	X	1.95	3.91	7.73	15.38
Ours+Int.	✓	X	**6.14**	**11.85**	**22.16**	**41.11**
Ours+Ext.	✓	✓	6.22	12.06	22.72	42.10
Amazon	**Sup**	**Ext**	**R@50**	**R@100**	**R@300**	**R@500**
WMD	X	X	**6.58**	**12.87**	**36.09**	**57.19**
Exemplar	X	X	3.34	6.11	15.59	23.93
Approx.EMD	X	X	5.13	9.77	26.60	39.76
Ours	X	X	**6.92**	**13.40**	**35.95**	**52.24**
SelfAttn+TL	✓	X	7.78	15.59	46.85	77.38
DSPN+TL	✓	X	2.29	4.53	13.40	22.17
Ours+Int.	✓	X	**7.88**	**15.76**	**47.21**	**78.18**
$\text{BERT}_{\text{BASE}}$	✓	✓	7.88	15.79	**47.50**	**78.87**
Ours+Ext.	✓	✓	**7.90**	**15.80**	47.34	78.65
Twitter	**Sup**	**Ext**	**R@50**	**R@100**	**R@200**	**R@300**
WMD	X	X	**5.90**	**11.62**	**22.89**	**33.99**
Exemplar	X	X	5.60	11.01	21.67	32.34
Approx.EMD	X	X	5.73	11.19	21.93	32.37
Ours	X	X	**5.78**	**11.35**	**22.00**	**32.58**
SelfAttn+TL	✓	X	5.71	11.27	22.36	33.42
DSPN+TL	✓	X	5.44	10.95	21.93	32.94
Ours+Int.	✓	X	**7.27**	**14.28**	**27.58**	**40.30**
$\text{BERT}_{\text{BASE}}$	✓	✓	6.95	13.80	27.17	**40.20**
Ours+Ext.	✓	✓	**7.21**	**14.14**	**27.30**	40.04

Similar to the previous experiment, we split the settings based on external training data. However, we choose only one competitor for the setting trained without external data, which is a general self-attention module with triplet loss (i.e., TL). This competitor equivalents to a method that directly learn a supervised classification task on small datasets without using transfer learning.

The results in Fig. 4 show the classification accuracy between two approaches that do not use external data (solid line) in which Ours+Int. approach outperforms TL at all situations.

For the approaches trained with external data, $\text{BERT}_{\text{BASE}}$ has surprisingly good performance even at a very low sample per class on BBCSports dataset. This is possibly due to the similarity between the target dataset and the pretrained data in $\text{BERT}_{\text{BASE}}$, which had been trained from BookCorpus (800M words) and English Wikipedia (2,500M words) datasets. However, Ours+Ext. achieves higher accuracy than $\text{BERT}_{\text{BASE}}$ for 30 samples per class or more for BBCSports and all settings for Twitter. This demonstrates that our approach has the capability to improve the accuracy of TL on a different target domain, e.g., Twitter in small dataset conditions.

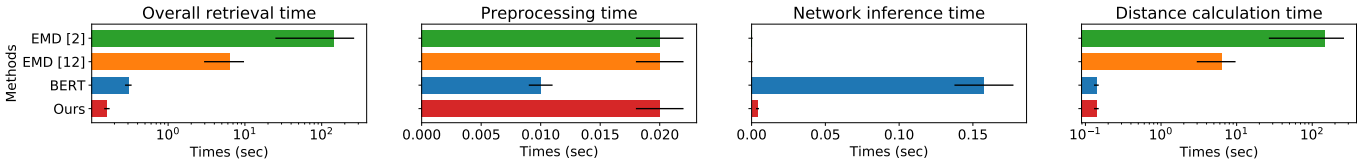

Fig. 5. Set of bar graphs shows an average time consumption for each steps in similarity search on Amazon dataset from a given query. Noted that we use log scale on an overall retrieval time and distance calculation time graph. Ours is the fastest approach in overall retrieval time and even faster than state-of-the-art neural network *BERT* with comparable accuracy.

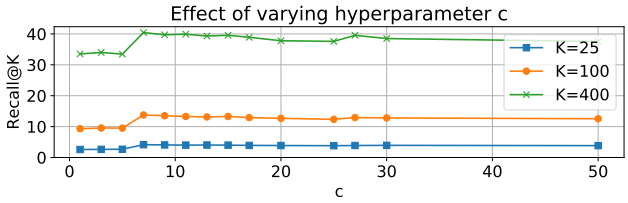

Fig. 6. Figure shows the effectiveness of our re-weighting negative terms on recall@K by tuning the hyperparameter c in Eq.3.

D. Experiments on computational efficiency

We evaluated the average query time on the Amazon dataset of our approach against classic EMD ($O(n^3 logn)$) [2], threshold EMD ($O(n^2 Ulogn)$) [12] and a neural network-based method, BERT. All of the studies were conducted on Intel Xeon E5-2698 v4 2.20GHz with 20-core multiprocessing for distance calculation and NVIDIA Tesla V100 for neural network inference.

Fig. 5 shows that our approach is the fastest method among the competitors. However, BERT uses a tokenizer to process raw text, which is faster than Word2Vec in other approaches. Nonetheless, BERT takes a longer time to encode a query to a feature vector than our model since our model has significantly fewer parameters of 4.9M while BERT has 110M parameters.

In the distance calculation phase, the neural network approaches (i.e., ours and BERT) which use Euclidean distance are *39.625 times* faster than Threshold EMD and *905.9 times* than classical EMD.

E. Hyperparameter study

We conducted a study to demonstrate the effectiveness of re-weighting negative terms by varying hyperparameters c in Eq.3 on Recipe, which is the most challenging dataset among all 4 datasets. Higher c means less effect of re-weighting. We observe that for all Recall@K, $c = 7$ to $c = 13$ have significantly better recall rates than higher c values as shown in Fig. 6. This shows the effectiveness of prioritizing high EMD negative pairs to force the network to learn with samples that are more likely to belong to different classes.

VII. Concluding Remarks

We propose a self-supervised deep metric learning approach for pointsets, which enables effective representation learning on unlabeled datasets. The key idea of our method is the use of EMD to generate pseudo labels for triplet loss training and a re-weighting technique that encourages larger separation between certain negative pairs deemed more likely to come from different classes. Our experiments show that our learned representation can be used for downstream tasks and produce results superior to other supervised competitors, including an NLP-specific approach based on a state-of-the-art language model. In addition, our method is shown to be the fastest among all the competitors. As future work, we plan to further generalize this approach to other pointsets data or other distance functions.

References

[1] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From word embeddings to document distances," in *ICML*, 2015.

[2] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *IJCV*, vol. 40, pp. 99–121, 2000.

[3] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," *CVPR*, pp. 4004–4012, 2016.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CVPR*, Jun 2015.

[5] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *CVPR*, pp. 4685–4694, 2018.

[6] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *NIPS*, 2016.

[7] D. Yao, G. Cong, C. Zhang, and J. Bi, "Computing trajectory similarity in linear time: A generic seed-guided neural metric learning approach," in *ICDE*, 2019, pp. 1358–1369.

[8] Z. Wang, C. Long, G. Cong, and C. Ju, "Effective and efficient sports play retrieval with deep representation learning," in *SIGKDD*, 2019.

[9] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. A. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *TPAMI*, vol. 38, pp. 1734–1747, 2016.

[10] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," *CVPR*, pp. 6203–6212, 2019.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[12] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *ICCV*, 2009, pp. 460–467.

[13] Y. Tang, L. H. U, Y. Cai, N. Mamoulis, and R. Cheng, "Earth mover's distance based similarity search at scale," *VLDB*, 2013.

[14] M. Kaya and H. Şakir Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, p. 1066, 2019.

[15] X. Cao, B.-C. Chen, and S.-N. Lim, "Unsupervised deep metric learning via auxiliary rotation loss," *ArXiv*, vol. abs/1911.07072, 2019.

[16] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," *CoRR*, vol. abs/1511.06391, 2016.

[17] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola, "Deep sets," in *NIPS*, 2017.

[18] Y. Zhang, J. S. Hare, and A. Prügel-Bennett, "Deep set prediction networks," in *NeurIPS*, 2019.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *ArXiv*, vol. abs/1706.03762, 2017.

[20] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *ICML*, 2018.

[21] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, "Transfer learning in natural language processing," in *ACL: Tutorials*, 2019.

[22] K. Skianis, G. Nikolentzos, S. Limnios, and M. Vazirgiannis, "Rep the set: Neural networks for learning set representations," *ArXiv*, vol. abs/1904.01962, 2019.